

지식분석도를 이용한 지식기반 웹 사이트 자동 생성 도구의 개발*

송 용 옥**, 김 우 주***, 홍 준 석****

Development of an Automatic Generation and Management Tool for Web-based Inference Sites

Yong Uk Song, Wooju Kim, June Seok Hong

Most of existing expert systems developed for Web use CGI-based techniques and this frequently makes them suffer from the overburden of commercial Web servers, which deal with large-scale services. However, since HTML-based inference technique represents expert's knowledge by hyperlinks among HTML documents, the hypertext function of the Web can perform the inference efficiently in terms of time and space without the help of additional inference engines. In spite of such benefits, when the expert's knowledge is relatively large and/or complicated, the HTML-based inference technique has usually become to have a hard time of dealing with a lot of HTML documents because generation and management tasks of the numerous HTML documents would cause big trouble to the knowledge engineer. To resolve this problem, we developed an automatic generation and management tool for Web-based inference sites, called WeBIS. With this tool, a knowledge engineer can input and edit expert's knowledge using Expert's Diagram on the GUI (Graphical User Interface) environment and automatically generate hyper-linked HTML documents for Web-based inference from the Expert's Diagram.

* 이 논문은 2001년도 연세대학교 학술연구비의 지원에 의하여 이루어진 것임.

** 연세대학교 경영·정보학부

*** 전북대학교 산업시스템공학과

**** 경기대학교 경영정보학전공

I. 서론

전자상거래 사이트가 단순한 상품 거래만의 기능을 넘어 고객 서비스를 통한 경쟁우위 확보도 중요시하게 됨에 따라, 웹 사이트에 기능을 부여하고자 하는 시도가 활발히 이루어지면서, 웹 기반 지능형 정보시스템 또는 전문가 시스템이 새로이 각광을 받고 있다. 현재까지 웹 환경에서 개발된 전문가 시스템의 응용분야는 지능형 전자메일 해석 및 분류(Intelligent E-Mail Interpretation and Classification), 제품 및 서비스에 대한 지능형 마케팅 자문 및 교육 시스템(Smart Advisor for Marketing Products & Services and Training), 온라인 구성 시스템(Online Configuration Systems), 기술지원 안내 시스템(Help Desk for Technical Support) 등이다.

그러나, 웹이 하이퍼 미디어(hyper-media)의 기능에 의해 편리한 사용자 인터페이스를 제공하고 시스템의 개발 및 유지보수를 쉽게 하고는 있으나, 웹의 무연결성(connectionless), 무상태성(stateless) 특성(Fielding *et al.*, 1998)은 사용자와의 여러 번에 걸친 대화를 필요로 하는 전문가 시스템을 구현하는 데에는 장애가 되고 있다. 이 문제를 해결하기 위하여 웹 기반 전문가 시스템들은 각 사용자별로 사실베이스를 구축하여 저장하고 있다. 하지만, 동시에 수십만, 수백만 명의 사용자들을 상대하여야 할 수도 있는 가혹한 웹 환경에서 이들 모두의 사실베이스를 유지하는 것은 서버의 메모리 자원에 심각한 부담이 될 수 있으며, 또한 특정 사용자의 사실베이스를 빠른 시간 내에 접근하기 위한 인덱싱(indexing) 작업 등의 추가적인 부담도 생기게 된다. 또한, 무거운 추론 엔진을 각 사용자별로 할당하기 위한 복잡한 트랜잭션 처리와 이에 따르는 시스템 자원의 고갈의 문제에도 직면하게 된다.

이에 대한 대안으로서 송용욱, 이재규[2000]는 HTML 기반 추론 기법을 제시하고, HTML 기반 추론 사이트 생성 프레임워크를 제안한 바 있다.

HTML 기반 추론은 질문 화면과 답변 화면이 서로 하이퍼링크(Hyperlink)된 HTML 파일들을 웹 서버를 통하여 서비스함으로써 별도의 추론 엔진 없이 웹 서버가 단지 웹 브라우저가 요청하는 HTML 파일만 제공하면 자연스럽게 추론이 진행되도록 하는 기법이다. 이 기법에서는 별도의 추론 엔진과 사용자별 사실베이스가 필요 없다. 사실베이스는 HTML 파일들 간의 하이퍼링크에 논리적으로 함축되어 있으며, 특정 사용자가 어느 HTML 파일을 보고 있는지가 그 사용자의 사실베이스의 내용을 결정한다. 또한, HTML 파일들 간의 하이퍼링크에 따라 요청(request)된 HTML 파일을 웹 서버가 자신의 하드디스크에서 읽어서 응답(response) 하기만 하면 되므로 별도의 추론 엔진이 필요 없게 된다. 그러므로, 일반 웹 기반 전문가 시스템들이 필요로 하는 복잡한 트랜잭션 처리 시스템이나 사실베이스를 저장하기 위한 큰 규모의 메모리 공간을 별도로 마련할 필요가 없으며, 따라서 아주 빠르고 효율적인 웹 기반 추론 서비스가 가능하게 된다.

그러나, HTML 기반 추론 기법은 서로 하이퍼링크(Hyperlink)된 수많은 HTML 파일들을 이용하는 것이기 때문에, 지식공학자가 수작업으로 지식을 HTML 파일들로 변환하고, 지식의 변화에 따라 그 파일들을 다시 수정하는 것이 매우 어렵다. 따라서, HTML 기반 추론에서는 이러한 HTML 파일들을 생성하고 관리하는 것이 중요한 문제로 지적되었다. 그러므로, 본 논문에서는 이 문제를 해결하기 위해 HTML 기반 추론용 웹 사이트를 생성하고 관리하기 위한 방법론을 제시하고, 이를 바탕으로 HTML 기반 전문가 시스템 개발 도구를 구현하고자 한다.

이 논문의 구성은 아래와 같다. 2장에서는 웹 기반 전문가 시스템의 구조들을 비교하면서 HTML 기반 추론의 장점과 필요성을 살펴보고, 3장에서 HTML 기반 추론 기법을 알아본다. 그리고, 4장에서는 지식공학자가 자신의 지식을 표현하는 도구로서 이용하고 있는 지식분석도를 HTML 기

반 추론의 관점에서 다시 생각해 본다. 이를 바탕으로 5장에서는 HTML 기반 추론 웹 사이트의 자동 생성 및 관리를 위해 개발된 WeBIS 시스템의 구조와 구현 방법론 및 기능 등을 설명한 후 6장에서 결론을 맺도록 하겠다.

II. 웹 기반 전문가 시스템의 구조 비교

송용욱, 이재규[2000]는 김평철[1996]의 웹과 데이터베이스의 연동 구조 분류를 바탕으로 웹

<표 1> 웹 기반 전문가 시스템의 구조 분류

대분류	소분류	내용 및 장단점
서 버 측	CGI	<ul style="list-style-type: none"> HTML의 FORM 태그(Tag)와 CGI 규약을 이용하여 사용자의 입력을 받아 처리 기존의 웹 서버, 웹 브라우저, URL, HTTP, HTML, CGI 등의 기술을 변경 없이 사용할 수 있으며, 시스템의 개발, 시험이 용이하며 추후의 시스템 확장과 유지보수가 쉬움 웹의 무연결성, 무상태성 특성 때문에 그 사용자와의 계속되는 대화의 내용을 기억하기 위한 방안(Session, Cookies 등)이 따로 필요 다수의 사용자가 동시에 서비스를 요구할 경우 시스템 자원의 부족과 성능저하를 초래 (이를 해결하기 위해 보통 light-weight dispatcher 사용) 기존의 대부분의 웹 기반 전문가 시스템들이 이 구조 이용
	웹 서버	<ul style="list-style-type: none"> 추론기관을 내장한 전문가 시스템 전용 웹 서버를 개발하거나 기존의 웹 서버에서 지원하는 확장 API(Application Program Interface; 예를 들어 Netscape의 NSAPI 등)를 이용하여 웹 서버에 추론기관을 추가 개발, 시험, 시스템 확장, 유지보수 등이 용이하지 않음 차후에 확장하거나 이식하고자 할 때 신축성이 떨어짐 이 구조를 갖는 웹 기반 전문가 시스템 사례는 없음
	HTML	<ul style="list-style-type: none"> HTML만을 사용 개발, 시험, 확장, 유지보수 등이 매우 용이하며, CGI를 사용할 때 나타나는 대규모 서비스 시의 시스템 부하 등의 문제가 없음 뒤로 가기 버튼에 의한 사실베이스의 혼란 문제가 없음 지식이 크거나 복잡할 경우 수 많은 HTML 문서들을 관리하는 것이 용이하지 않음 HTML만을 사용하여 얼마나 일반적인 전문가 시스템을 구현할 수 있는가 하는 것이 문제
클 라 이 언 트 측	외부 뷰어	<ul style="list-style-type: none"> 웹 브라우저가 자체적으로 처리할 수 없는 메시지에 대하여 자동으로 수행시키는 외부 프로그램을 이용 추론기관은 전적으로 사용자의 컴퓨터에서 별도의 응용 프로그램으로서 수행되기 때문에 서버와 네트워크에 대한 추가 부담이 없으며, 따라서, 서버측에 추론기관을 둔 경우 보다 빠르게 추론 결과를 얻을 수 있음. 또한, 웹의 무연결성, 무상태성 특성에서 오는 문제들도 나타나지 않음 사용자가 웹 브라우저 이외에 외부 뷰어라는 별도의 소프트웨어의 기능을 익혀야 하며, 또 외부 뷰어를 사용하기 위하여 사전에 이것을 다운로드를 받아서 설치하여야 함 외부 뷰어는 외부 뷰어가 설치되어 수행될 시스템 플랫폼에 맞추어 여러 가지 버전이 준비되어야 함. 또, 외부 뷰어 프로그램이 변경되면 모든 사용자는 변경된 소프트웨어를 다시 다운로드 받아 설치하여야 하므로 시스템 확장과 유지보수가 어려움 웹 출현 이전의 전통적 전문가 시스템을 수정 없이 그대로 사용하는 형태
	웹 브라우저	<ul style="list-style-type: none"> Java Applet[Lemay and Cadenhead, 1998]을 이용 외부 뷰어에서 나타나는 시스템 종속성, 설치 등의 모든 문제를 해결, 추론기관의 개발, 시험, 확장, 유지보수 용이, 외부 뷰어의 장점을 그대로 가짐 웹 서버로부터 Java 바이트코드를 전달 받는 시간이 긴 것이 문제

기반 전문가 시스템 구조들을 분류하고 장단점을 비교한 바 있다. 그에 따르면, 웹 환경에서의 전문가 시스템은 추론기관이 서버(Server) 측에 있는 경우와 클라이언트(Client) 측에 있는 경우로 대별된다. 추론기관이 서버 측에 있는 경우는 다시 세 가지로 나뉘어진다. 첫째는 CGI를 이용하는 경우이고, 둘째는 웹 서버 자체가 추론기관을 내장하는 경우이고, 셋째는 HTML 만을 이용하는 경우이다. 그리고, 추론기관이 클라이언트 측에 있는 경우는 다시 두 가지로 나뉘어지는데, 첫째는 외부 뷰어(External Viewer)를 이용하는 경우이고, 둘째는 웹 브라우저에 추론기관을 포함시키는 경우이다. 이들의 내용과 장단점은 <표 1>에 정리되어 있다.

현재 일반적으로 많이 사용되는 방식은 CGI 방식이며, 그 이유는 CGI 방식이 WWW의 장점을 가장 많이 살리는 방식이기 때문인 것으로 보인다(http://www.exsys.com/, http://www.blazesoft.com/, http://www.ilog.com/). 그러나, 동시 사용자의 수가 많아질수록 서버의 부담이 커지는 점이 이 방식의 가장 큰 문제로 지적되며, 이를 극복하기 위하여 프로세스의 수를 줄이는 FastCGI, Multi-thread 등의 기법[최환진, 1999]이 사용되기도 하며, 또한 수 많은 클라이언트의 요구를 우선순위에 따라 관리하기 위한 트랜잭션 처리 기법도 사용된다. ASP, PHP, Java Servlet, JSP 등이 이러한 기법들을 적용한 CGI 방식의 발전형태로 볼 수 있다[최환진, 1999]. 그러나, 웹의 무연결성, 무상태성 특성은 세션 유지와 별도의 사실베이스를 필요로 함과 동시에, 전문가 시스템이 사용자와 대화를 할 때마다 처음부터 추론을 다시 하여야 하는 문제는 여전히 남게 된다. 또한, 사실베이스가 서버 측에서 운영될 경우, 사용자가 웹 브라우저의 “뒤로 가기” 버튼을 눌러 답변 내용을 “취소”하였을 때 사실베이스에 혼란이 오는 문제를 해결하는 것도 주요한 과제 중의 하나이다.

이에 반하여, HTML 방식이 트랜잭션의 처리,

사실베이스 유지 등에 따르는 웹 서버 부하의 문제, 재추론의 문제, “뒤로 가기” 버튼에 의한 사실베이스 혼란의 문제 등을 해결하였음에도 불구하고, CGI나 Java Applet 등의 방식과는 대조적으로 HTML 방식에 대한 관심은 상대적으로 매우 적었던 것으로 보인다. 그러나, HTML의 하이퍼링크 기능에 의해 추론을 진행할 수 있다는 점에 착안하면, 추론을 이용한 진단, 분류 등의 분석문제에 관한 HTML 방식의 전문가 시스템은 충분히 구현 가능하며, 이렇게 구현된 시스템은 <표 1>에서와 같은 장점을 갖게 된다. 따라서, 본 논문에서는 이러한 HTML 기반 추론을 현실화할 수 있는 방안으로서 HTML 기반 추론 웹 사이트 생성 및 관리 도구를 개발하고자 하며, 이를 위한 이론적 배경으로서 HTML 기반 추론에 대하여 다음 절에서 설명하도록 한다.

Ⅲ. HTML 기반 추론

송용욱, 이재규[2000]는 웹의 하이퍼링크 기능을 이용하여 별도의 추론 엔진이 없이 웹 상에서 HTML만으로 추론 사이트를 제공할 수 있음을 보인 바 있다. 예를 들어 <그림 1>과 같은 규칙베이스가 있다고 하자.

```
IF 신호등 IS 빨간색
THEN 서시오

IF 신호등 IS 파란색
THEN 가시오
```

<그림 1> 건물목 예제 규칙베이스

이때 <그림 2>와 같은 세 개의 HTML 파일(a.html, b.html, c.html)로 <그림 3>과 같은 웹 사이트를 구성하면, 위의 규칙베이스에 대해 추론을 수행해주는 웹 사이트가 구현된다. <그림 3>의

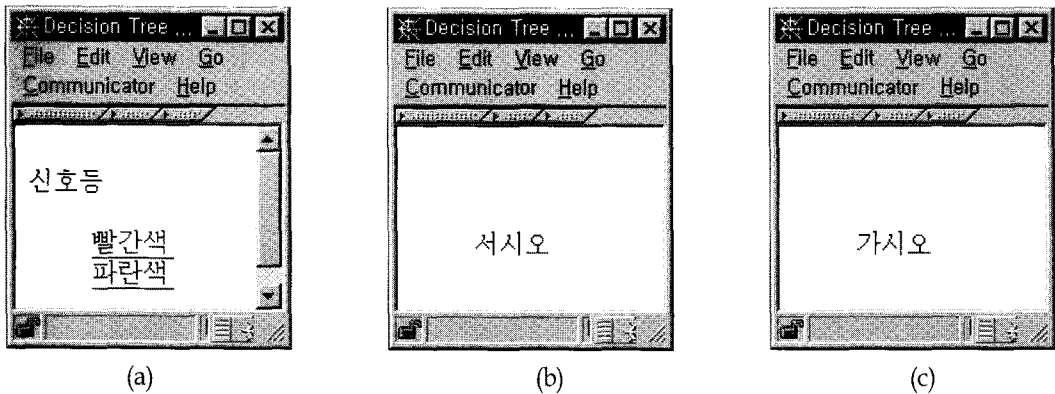
파일: a.html	
<pre> <HTML> <HEAD> <TITLE> Decision Tree </TITLE> </HEAD> <BODY>
 신호등

 <BLOCKQUOTE> 빨간색
 파란색 </BLOCKQUOTE> </BODY> </HTML> </pre>	
파일: b.html	파일: c.html
<pre> <HTML> <HEAD> <TITLE> Decision Tree </TITLE></HEAD> <BODY>

 <BLOCKQUOTE> 서시오 </BLOCKQUOTE> </BODY> </HTML> </pre>	<pre> <HTML> <HEAD> <TITLE> Decision Tree </TITLE></HEAD> <BODY>

 <BLOCKQUOTE> 가시오 </BLOCKQUOTE> </BODY> </HTML> </pre>

<그림 2> 추론을 위한 HTML 파일

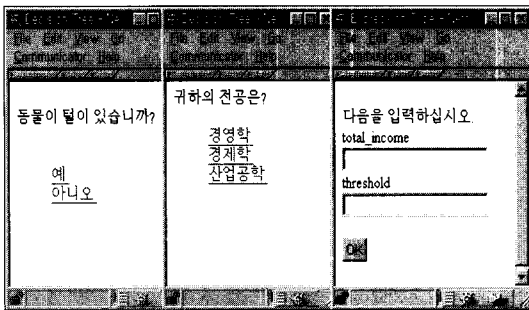


<그림 3> HTML 기반 추론 사이트

(a)에서 하이퍼텍스트(Hypertext) “빨간색”은 <그림 3>의 (b)에 하이퍼링크 되어 있고, 하이퍼텍스트 “파란색”은 <그림 3>의 (c)에 하이퍼링크 되어 있다. 이때, 사용자가 화면 <그림 3> (a)의 “신호등” 질문에 대해 “빨간색”을 클릭하면 “서시오”라는 결론을 보여주고, “파란색”을 클릭하면 “가시오”라는 결론을 보여줌으로써 <그림 1>의 규칙베이스에 대한 추론이 수행되게 된다.

추론을 구현하기 위해서는 사실(Fact)의 입력, 하위목표(Subgoal)에 의한 연쇄, 추론결과의 출력 등이 지원되어야 한다[이재규 외, 1996A]. HTML 기반 추론의 경우, 하위목표에 의한 연쇄는 위에서 이야기한 바와 같이 하이퍼링크에 의해 구현할 수 있으며, 추론결과의 출력은 웹 브라우저에 결론을 디스플레이 함으로써 이루어진다. 한 가지 남은 문제는 사실의 입력을 웹 상에

서 지원하는 방안이다. 추론에서 사실 값은 변수에 저장된다. 추론에서 사용하는 변수의 형태는 진위형(Fact Type), OAV(Object-Attribute-Value Type)형, 수치형(Numeric Type)의 세 가지가 있다[이재규 외, 1996A; 1996B]. 여기서, 진위형과 OAV형 변수 값의 입력은 앞서 설명한 하이퍼텍스트의 나열에 의해 쉽게 해결된다. 예를 들어 <그림 1>의 규칙베이스 예제에서 “신호등” 변수는 “빨간색” 또는 “파란색”을 값으로 갖는 OAV 형 변수로서 이것은 <그림 3>의 (a)처럼 구현되고, 사용자의 “빨간색” 또는 “파란색”에 대한 마우스 클릭이 이 변수의 값을 입력하는 행위가 된다. 진위형 변수의 경우도 “참”과 “거짓”을 나열하면 된다. 예를 들어 <그림 4>의 첫 번째 화면은 진위형 변수가 나타난 규칙의 조건절을 구현한 것이다.



<그림 4> 변수 형태별 질문 화면

수치형 변수의 경우는 값을 입력 받은 후 수식 계산이 이루어져야 하기 때문에 하이퍼텍스트의 단순 나열로는 해결되지 않는다. 이를 해결하기 위한 방법은 JavaScript[Danesh, 1996], VBScript[Jerke et al., 1997] 등과 같은 클라이언트 측 스크립트 언어(Client-side Script Language)¹⁾를 이용하는 것이다. 예를 들어 다음과 같

은 규칙이 있다고 하자.

```
IF total_income >= 0.2 * threshold
THEN pay_tax
```

```
IF total_income < 0.2 * threshold
THEN do_not_pay_tax
```

위 두 규칙에서 total_income과 threshold 변수는 수치형 변수이다. 이 두 규칙은 <그림 5>와 같이 JavaScript를 사용하여 HTML 파일에 의해 구현될 수 있다. <그림 5>에서 보듯이 규칙의 조건절에 나타나는 모든 변수들의 값은 FORM 태그에서 입력 받으며, 입력된 변수를 포함한 수식의 계산 및 이에 따른 하이퍼링크는 SCRIPT 태그 내의 “Branch” 함수에서 이루어진다. JavaScript 자체가 하나의 프로그래밍 언어이므로 조건절의 수식이 프로그래밍 언어의 if 문에서 사용되는 임의의 모든 수식이 될 수 있다는 것이 이 방식의 또 다른 장점이다. 수치형 변수에 대하여 질문화면을 구성한 예는 <그림 4>의 세 번째 화면에 나타나 있다.

HTML 기반 추론 기법은 웹 서버의 부담이 없고 속도가 빠르며, 별도의 추론 엔진을 구현할 필요가 없으며, HTML에 대한 기본적인 지식만 있는 사람이면 누구든지 웹 기반 추론 사이트를 만들 수 있다는 점에서 단순하면서도 강력한 웹 기반 추론 기법이라고 할 수 있다. 다만, 이 기법을 사용할 경우 문제가 되는 것은 전문가의 지식을 표현한 HTML 파일들이 많아질 경우 그들 간의 하이퍼링크를 수작업에 의해 관리하는 것이 매우 어렵게 된다는 점이다. 다음 절부터는 이 문제를 해결하기 위한 방안을 알아본다.

1) Script 언어란 컴파일러가 아닌 해석기(Interpreter)에 의해 해석되면서 바로 수행되는 프로그래밍 언어를 말한다. 웹 환경에서는 웹 서버 측에서 해석

되어 수행되는 스크립트 언어와 웹 브라우저 측에서 해석되어 수행되는 스크립트 언어가 있다. JavaScript와 VBScript는 웹 브라우저 내의 해석기에 의해 수행되어 웹 브라우저에 결과를 표시하는 대표적인 클라이언트 측 스크립트 언어이다.

```

<HTML>
<HEAD>
  <TITLE> Expression Type </TITLE>
  <SCRIPT LANGUAGE = "JavaScript">
    <!--
      function Branch(form)
      {
        if (form.total_income.value >= 0.2 *
            form.threshold.value)
          location = "http://iis.kaist.ac.kr/tax.
            html"
        else
          location = "http://iis.kaist.ac.kr/notax.
            html"
      }
    //-->
  </SCRIPT>
</HEAD>
<BODY>
<FORM NAME = "exprform">
다음을 입력하십시오. <BR>
total_income <INPUT TYPE = "text"
  NAME = "total_income"> <BR>
threshold <INPUT TYPE = "text" NAME =
  "threshold"> <BR>
<INPUT TYPE = "button" VALUE = "OK"
  onClick = "Branch(this.form)">
</FORM>
</BODY>
</HTML>

```

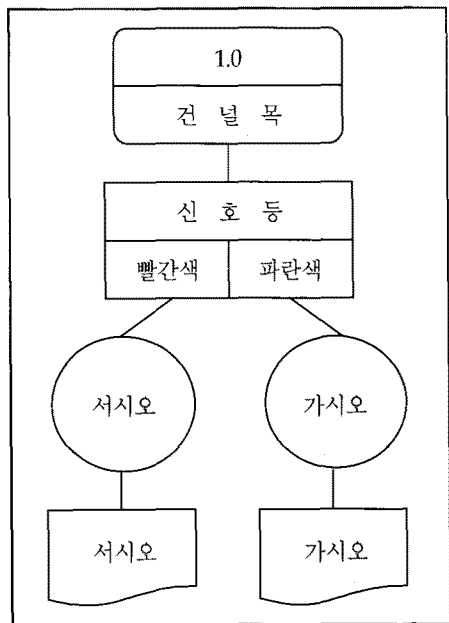
<그림 5> JavaScript를 이용한 수치형 변수의 구현

IV. 지식분석도

HTML과 JavaScript를 이용하여 구현된 추론 웹 사이트의 문제점으로 지적할 수 있는 것이 웹 사이트 관리의 문제이다. 이러한 웹 사이트는 수 많은 HTML 파일 및 그림 파일들로 이루어져 있을 것이고, 따라서 이들을 수작업으로 관리, 변경하는 것은 규칙의 수가 많아지고 또 규칙이 복잡해질수록 더욱 더 어려워질 것이다. 일반 웹 사이트의 경우에도 웹 사이트 저작 및 관

리용 도구들이 수많이 개발되어 상품화되고 있는 것을 보더라도, 웹 사이트의 생성 및 관리는 상당히 손이 가는 작업이다. 따라서, HTML 기반 추론 웹 사이트의 경우에도 그러한 저작 및 관리 도구가 더욱 더 필요해진다.

J.K. Lee 등[Lee, Lee, and Choi, 1990; 이재규 외, 1996A; 이재규 외, 1996B]은 전문가의 지식을 지식분석도(Expert's Diagram)라는 그래프로 표현 한 후 이 그래프로부터 추론기관용 규칙을 생성하는 방안을 제시한 바 있다. 이 지식분석도는 원래 인간 전문가로부터의 지식획득을 용이하게 하고, 지식의 유지보수를 도와주기 위한 도구로서 개발된 것이므로, HTML 기반 추론 웹 사이트를 생성하고 관리할 때도 유용하게 사용할 수 있다. 예를 들어 <그림 1>에 표현된 규칙베이스를 지식분석도로 표시하면 <그림 6>과 같다. <그림 6>에서 시작 노드(Topic Node)와 결론노드(Conclusion Node)를 제외한 중간노드(Branch Node) 1개와 단말노드(Display Node) 2개가 각각 하나의 HTML 파일에 대응되며, 이



<그림 6> 지식분석도

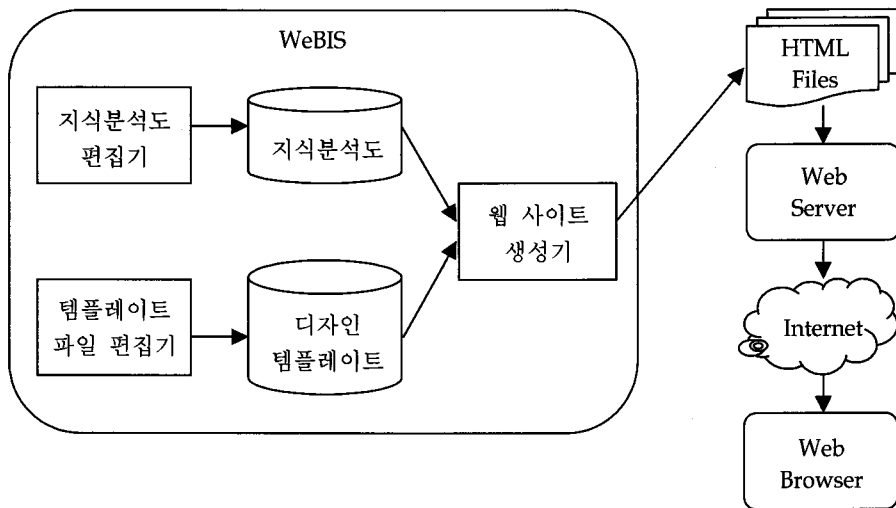
것은 각각 <그림 3>의 a.html, b.html, c.html에 해당된다. 따라서, 지식분석도가 주어지면 지식 분석도의 노드들을 각각 하나의 HTML 파일로 변환해 주는 시스템의 구현이 가능하다.

원래 지식분석도는 현장 전문가로부터의 지식 습득을 그래프에 의해 지원하기 위하여 개발된 것이다. 규칙이라는 개념에 익숙하지 못한 현장 전문가로 하여금 자신의 지식을 그래프로 표현하도록 함으로써 지식 습득을 용이하게 한 것이었다. 그리고, 지식분석도를 시스템으로 구현한 LIFT의 경우, 지식분석도로 표현된 지식을 SKI2라는 역방향 추론기관용 규칙베이스로 자동 변환해 주는 기능을 갖고 있었다. 이 개념의 연장선 상에서 보면, 지식분석도로 표현된 지식으로부터 웹 기반 추론을 위한 HTML 파일들의 자동생성 역시 가능하다. 지식분석도를 지원하는 시스템이 개발된다면 그래프 지원에 의한 현장 전문가로부터의 지식습득이 용이해짐과 동시에 지식이 변경되었을 때 지식분석도를 변경한 후 변경된 지식분석도로부터 HTML 파일들을 자동 생성하면 되므로 지식베이스의 유지보수가 쉬어지게 된다. 또한, 이것은 동시에 HTML 기반 추론용 웹 사이트의 생성 및 관리가 용이해

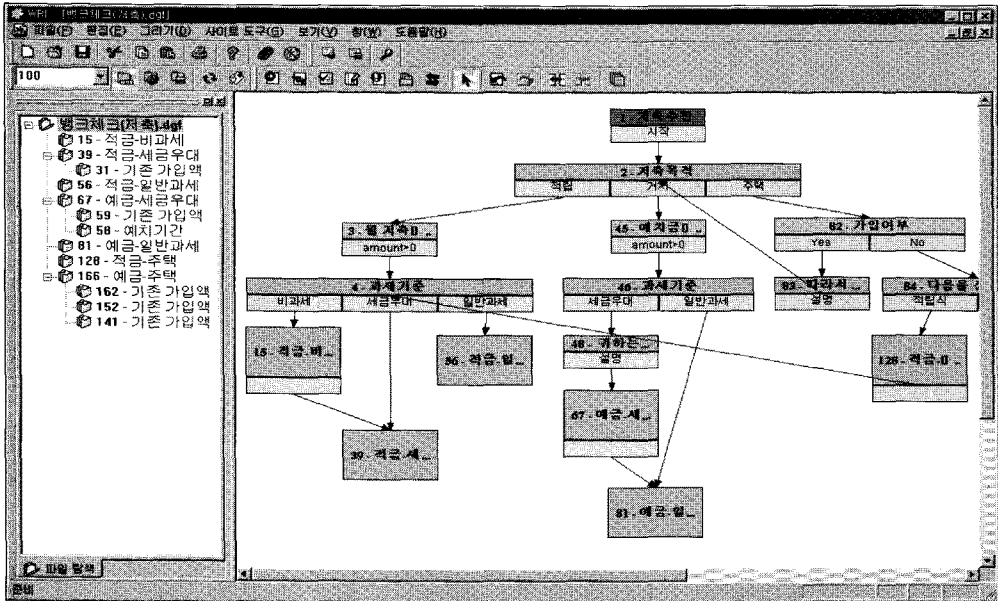
진다는 의미이기도 하다. 다음 장에서는 추론사이트의 자동생성 및 관리를 위해 지식분석도를 입력받고, 입력된 지식분석도로부터 HTML 파일들을 자동 생성해주는 시스템에 대하여 설명한다.

V. 웹 기반 추론 사이트 자동 생성 도구

웹 기반 추론 사이트 자동 생성 도구인 WeBIS (Web-based Inference System)와 이를 이용한 웹 기반 추론시스템의 구조는 <그림 7>에 나타나 있다. WeBIS의 사용자인 지식공학자(Knowledge Engineer)는 WeBIS의 지식분석도 편집기를 이용하여 지식분석도를 입력한다. 앞에서 설명한 바와 같이 지식분석도는 규칙형태의 지식을 그래프 구조로 표현한 것이다. 입력된 지식분석도로부터 WeBIS의 웹 사이트 생성기는 클라이언트 측 스크립트 프로그램을 포함한 HTML 파일들을 생성한다. 이 HTML 파일들은 입력된 지식에 따라 추론이 진행될 수 있도록 서로 하이퍼링크 되어 있다. 생성된 HTML 파일들은 웹 서버에 등록되어 인터넷을 통해 서비스된다.



<그림 7> WeBIS 및 웹 기반 추론 시스템의 전체 구성도



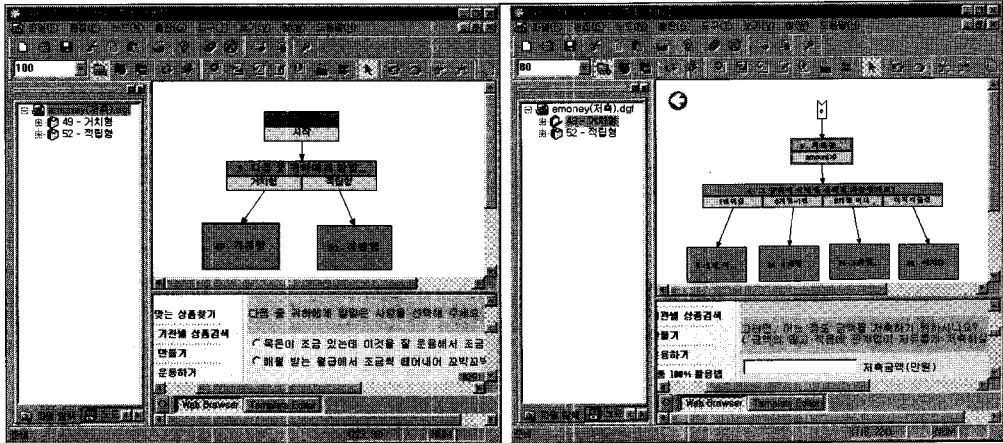
<그림 8> 지식분석도 입력 화면

HTML 파일 생성시, 생성되는 HTML 파일의 화면 디자인 또한 고려해야 할 요소이다. WeBIS에서는 지식과 화면 디자인을 분리하기 위하여 디자인 템플레이트(Template)를 사용한다. 디자인 템플레이트는 사용자에게 보여줄 화면상의 내용들은 HTML로 표현하고, 추론을 위한 하이퍼링크가 표현될 부분은 특정한 구분자(delimiter)를 사용하여 표시한 파일로서, WeBIS가 웹 페이지들을 생성할 때 디자인 부분은 그대로 사용하고, 구분자로 표시된 부분만 하이퍼링크를 포함하는 문장으로 바꿀 수 있도록 되어있다. 템플레이트 파일 편집기는 나모 웹 에디터, MS 프론트 페이지 등 일반적인 웹 페이지 편집기를 사용할 수 있도록 WeBIS가 지원하므로, 이 절에서는 WeBIS에서의 지식분석도 편집기와 디자인 템플레이트를 이용한 웹 사이트 생성기에 대하여 설명한다.

5.1 지식분석도의 입력 및 편집

WeBIS를 이용한 금융상품 추천분야에 대한 지

식분석도 입력화면 예제가 <그림 8>에 나타나 있다. 화면 우측의 그래프가 지식분석도이며, 지식분석도는 노드(node)와 호(arc)로 이루어져 있다. 지식분석도 노드의 종류는 시작 노드, 진위형 노드, 선택형 노드, 수치형 노드, 결론 노드 및 그룹 노드가 있다. 시작 노드는 추론의 시작 점을 표시하며, 생성되는 추론 사이트의 첫 시작 페이지를 생성하기 위한 것이다. 진위형 노드는 진위형 변수를, 선택형 노드는 OAV형 변수를, 수치형 노드는 수치형 변수를 위한 노드로서 노드의 하단부에는 분기 조건을 입력하도록 되어 있다. 예를 들어, <그림 8>의 "4. 과세기준" 노드는 선택형 노드이며, 그 분기 조건으로 "비과세," "세금우대," "일반과세"의 세 가지를 갖고 있다. 다시 말하면, "과세기준"이 "비과세"인 경우, "세금우대"인 경우, "일반과세"인 경우 등 각 조건 중 하나가 만족되면 그 조건에 해당하는 호로 연결된 다음 노드로 이행하여 그 다음 질문을 진행하도록 하는 것이다. 결론 노드는 여러 질문 노드의 만족되는 각 조건들을 거쳐 결론이 이루어졌을 때, 그 결론을 표시하기 위한 노드이다.



<그림 9> Levels of Abstraction 지원

지식분석도를 입력할 때 노드의 수가 증가하게 되면 그림의 크기가 커지고 각 노드 간의 연결관계가 복잡해짐으로써 지식공학자가 지식분석도를 일목요연하게 파악하여 지식을 추가하거나 수정하는 것이 어렵게 된다. 이러한 경우에 다수의 노드를 포함한 지식분석도를 간략한 형태로 관리할 수 있도록 WeBIS는 Levels of abstraction을 지원한다. 즉, 하나의 지식분석도를 여러 단계의 상위 레벨과 하위 레벨의 그림들로 나누어 표현할 수 있도록 지원한다. 이를 위하여 도입된 노드가 그룹 노드이다. 그룹 노드는 여러 개의 (하위) 노드들을 하나로 묶어서 만든 것으로 하위 노드가 될 수 있는 노드에는 그룹 노드도 포함되므로 결과적으로 여러 레벨의 지식분석도를 그리는 것이 가능하게 된다. 예를 들어, <그림 9>의 좌측 화면의 “49. 가치형” 그룹 노드의 하위 레벨 지식분석도가 <그림 9>의 우측 화면에 나타나 있다.

5.2 디자인 템플레이트를 이용한 웹 사이트 생성

지식분석도는 추론을 위한 지식을 의사결정 그래프(decision graph)의 형태로 표현한다. 그리고, 지식분석도의 그룹 노드를 제외한 각 노드

는 각각 하나의 웹 페이지(= HTML 파일)로 생성된다. 웹 페이지의 디자인 요소는 웹 사이트의 사용자 인터페이스를 위해 간과할 수 없는 부분이다. 그러나, 웹 페이지의 디자인 부분을 HTML을 이용하여 지식분석도의 각 노드마다 별도로 입력하는 것은 매우 단순하면서도 시간을 소모하는 작업일 것이다. 따라서, WeBIS에서는 디자인 부분을 분리하여 디자인 템플레이트에 표시하고, 지식분석도의 각 노드에는 디자인 템플레이트 파일의 이름과 생성될 HTML 파일의 이름만 입력하도록 함으로써 위와 같은 문제를 해결하였다.

<그림 10>에 디자인 템플레이트 파일의 예제가 나타나 있다. 기본적으로 디자인 템플레이트 파일은 HTML 규약에 따라 작성하며, 지식분석도로부터 생성되어야 할 하이퍼링크와 스크립트 프로그램 등의 요소들에 대해서는 특별한 구분자(delimiter)와 명령어를 사용하여 그것들이 나중에 생성될 위치만 표시한다. <그림 10>에서 “<!-- ... -->”로 표시된 부분들이 그 구분자와 명령어들이다. 예를 들어 <!-- WGBRANCH --> 부분은 하이퍼링크를 위한 스크립트 프로그램이 들어갈 부분이며, <!-- WGINPUTFORM --> 부분은 진위형, OAV형, 수치형 등 질문 변수의 유형에 따라 질문 및 입력을 위한 내용이 들어갈 부

분이다. 특히, 구분자 “<!-- ... -->”는 HTML 관점에서는 주석문에 해당되므로, 나모 웹 에디터, MS 프론트 페이지 등 일반 웹 페이지 편집기를 사용하여 디자인 템플레이트 파일을 편집할 경우 WeBIS의 명령어들은 주석으로 처리되어 화면에 보이지않고 다른 HTML 부분만 보이므로, 기존의 웹 페이지 편집기를 이용한 디자인 템플레이트 파일의 편집이 용이하게 된다. WeBIS는 지식공학자가 자신이 원하는 웹 페이지 편집기를 지정하여 디자인 템플레이트 파일 편집에 이용할 수 있도록 지원하고 있다.

```

<HTML>
<HEAD>
<TITLE> <!-- WGITITLE --> </TITLE>
<!-- WGBRANCH -->
</HEAD>
<BODY>
<!-- WGINPUTFORM -->
</BODY>
</HTML>
    
```

<그림 10> 디자인 템플레이트 파일

5.2.1 선택형 노드

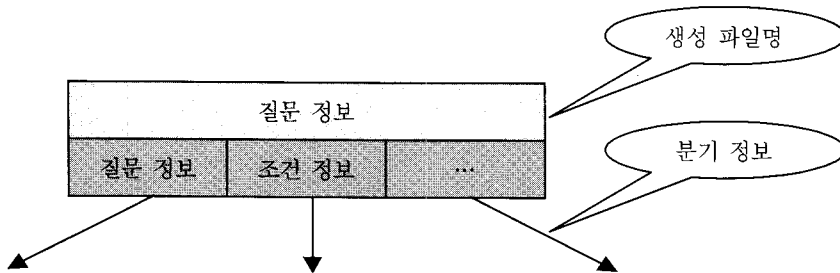
디자인 템플레이트를 이용한 웹 사이트 생성 방법을 설명하기 위하여, 먼저, 선택형 노드로부터 HTML 파일을 생성하는 방법을 설명하도록 한다. 선택형 노드에 표현되는 정보는 질문, 조

건, 분기, 생성 파일명 등이다. 이 정보들은 지식 분석도에서 노드, 호 및 노드 속성 등에 표시된다. 지식분석도에서의 정보 표시 위치가 <그림 11>에 도시되어 있다. 질문 정보는 노드의 상위 박스에 표시되고, 조건 정보는 노드의 하위 박스들에 표시되며, 분기 정보는 노드의 하위 박스와 다른 노드 간을 연결하는 호에 표시된다. 생성 파일명은 노드의 속성 정보로서 설정된다.

예를 들어 <그림 8>의 “4. 과세기준” 노드를 다시 한번 살펴보면, 질문은 “과세기준”이고, 조건은 “비과세,” “세금우대” 및 “일반과세”이다. 또한 조건 “비과세”는 “15. 적금 - 비...”으로 분기되고, “세금우대”는 “39. 적금 - 세...”으로 분기되며, “일반과세”는 “56. 적금 - 일...”으로 분기된다. 또한, 노드의 속성정보로 이 노드가 HTML 파일로 생성될 때 부여될 생성 파일명이 설정되어 있다. 이 파일명은 지식공학자가 지정할 수도 있고, 지정하지 않았을 경우에는 WeBIS가 자동으로 일련번호를 부여한다. 예제에서 “4. 과세기준” 노드에 부여된 생성 파일명을 “e_s4.html”이라고 하자. 또한, “비과세,” “세금우대” 및 “일반과세” 등의 조건별 분기에 의해 연결된 노드들에 부여된 생성 파일명을 각각 “e_s31.html,” “e_s34.html,” “e_s35.html”이라고 하자. 그러면, “비과세” 조건에 대해서는 다음과 같은 하이퍼링크가 생성된다.

```

<A HREF = “e_s31.html”> 비과세 </A>
    
```



<그림 11> 지식분석도에 표시되는 정보

디자인 템플레이트 파일	파일명 : e_s4.html
<pre> <HTML> <HEAD> <TITLE> <!-- WGITITLE --> </TITLE> <!-- WGBRANCH --> </HEAD> <BODY> <TABLE> <TR> <TD> </TD> <TD> <!-- WGINPUTFORM --> </TD> </TR> </TABLE> </BODY> </HTML> </pre>	<pre> <HTML> <HEAD> <TITLE> 과세기준 </TITLE> </HEAD> <BODY> <TABLE> <TR> <TD> </TD> <TD> 과세기준
 비과세
 세금우대
 일반과세
 </TD> </TR> </TABLE> </BODY> </HTML> </pre>

<그림 12> 선택형 노드에 대한 디자인 템플레이트와 생성된 HTML 파일

의미론적으로 이 A(Anchor) 태그가 나타내는 바는, 이 전문가 시스템의 사용자가 “과세기준”이라는 질문에 대해서 “비과세”라고 대답(클릭) 하였을 때 “e_s31.html”로 하이퍼링크가 일어나므로써 “비과세”라는 조건이 만족된 경우에 해당하는 다음 질문 또는 결론 화면이 사용자에게 보여지게 된다는 것이다.

이와 같은 방법으로 <그림 8>의 “4. 과세기준” 노드에 대하여 디자인 템플레이트 파일(<그림 12> 좌측)을 적용하여 생성한 HTML 파일과 그 디스플레이 화면이 <그림 12> 우측 및 <그림 13>에 나타나 있다. <그림 12> 우측의 생성된 파일명은 “4. 과세기준” 노드에 설정된 생성 파일명인 “e_s4.html”이다. “<!-- WGINPUTFORM -->” 명령어 부분에는 질문(“과세기준”)과 대답(“비과세,” “세금우대,” “일반과세”)이 나타나 있으며, 대답에 따라 하이퍼링크 될 내용들이 A 태그로 표시되어 있다. 한편, “<!-- WGBRANCH -->” 명령어 부분은 수치형 노드를 처리하기 위한 클라이언트 측 스크립트 프로그램이 들어갈 부분으로서 선택형 노드와는 관련이 없으므로 아무

것도 생성되지 않아 공백으로 남아있다. 진위형 변수의 경우에도 선택형 변수와 동일한 방법으로 생성이 가능하므로 그에 대한 설명은 생략한다.

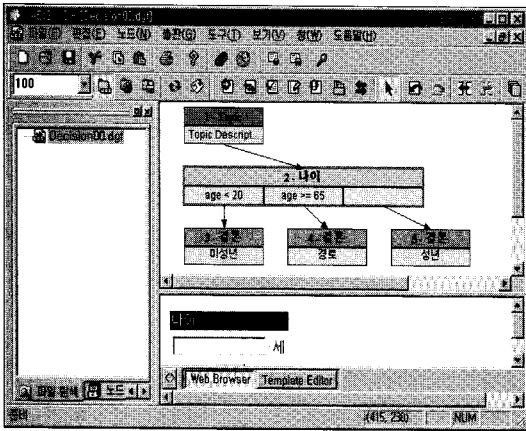


<그림 13> 선택형 노드의 디스플레이 화면

5.2.2 수치형 노드

선택형 노드와 마찬가지로 수치형 노드에 표현되는 정보도 질문, 조건, 분기, 생성 파일명 등이다. 이 중에서 질문, 분기, 생성 파일명은 선택

형 노드의 경우와 동일하나 조건은 문자열이 아닌 수식의 형태를 띠게 되므로, 수식에 대한 정보와 함께 수식에서 사용되는 변수에 대한 정보가 필요하다. 예를 들어, <그림 14>의 “2. 나이” 노드의 조건으로는 “age < 20,” “age >= 65” 등이 있으며, 이들 수식에서 사용하는 변수로는 “age”가 있다. 변수에 대한 정보는 변수의 이름과 변수에 대한 질문 관련 사항 등이 있으며, 이것들은 지식공학자가 입력하도록 되어 있다.



<그림 14> 수치형 노드 예제

수치형 노드의 경우에는 질문, 조건, 분기, 생성 파일명 등의 정보로부터 HTML 파일의 입력 부분 및 클라이언트 측 스크립트 프로그램이 생성되어야 한다. 입력 부분은 변수에 대한 정보로부터 생성된다. 예를 들어, “age” 변수의 이름은 “age”이고 질문 관련 사항이 “세”(歲)라고 할 때 “age” 변수에 대한 입력 부분으로 다음과 같은 HTML 문장이 생성된다.

```
<INPUT TYPE = “text” NAME = “age > 세
```

위 입력 양식이 웹 브라우저에 나타나는 모습은 <그림 14>의 우측 하단에 나타나 있다. 물론, INPUT 태그는 FORM 태그 내에서 사용되는 것이므로 입력 부분을 완성하기 위해서는 다음과

같은 완전한 FORM 문장이 생성되어야 한다.

```
<FORM METHOD = “get” NAME = “WBIForm1”>
<TABLE>
  <TR> <TD> 나이 </TD> </TR>
  <TR> <TD> <INPUT TYPE = “text” NAME =
    “age”> 세 </TD> </TR>
  <TR> <TD> <INPUT TYPE=BUTTON VALUE =
    “확인” OnClick = “WBIBranch
    (this.form)”> </TD> </TR>
</TABLE>
</FORM>
```

클라이언트 측 스크립트 프로그램은 위에서 생성된 FORM 문장 중 INPUT 태그의 NAME 어트리뷰트 정보와 분기 정보 및 생성 파일명 정보들을 이용하여 생성된다. 분기 조건으로 “age < 20,” “age >= 65” 및 빈(null) 조건이 있고, 하이퍼링크 될 파일들이 각 조건별로 “Decision3.html,” “Decision4.html,” “Decision5.html”이라고 할 때 다음과 같은 JavaScript 프로그램이 생성된다.

```
function WBIBranch(form)
{
  if (form.age.value < 20)
    location = “Decision3.html”;
  else if (form.age.value >= 65)
    location = “Decision4.html”;
  else
    location = “Decision5.html”;
}
```

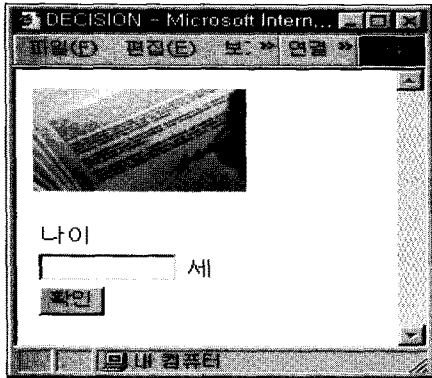
위 프로그램에서 함수명 WBIBranch는 입력 부분 “확인” 버튼(<INPUT TYPE = BUTTON VALUE = “확인” OnClick = “WBIBranch(this.form)”>)의 OnClick 이벤트와 연결되어 있고, form.age.value라는 변수명은 NAME 어트리뷰

<pre> <HTML> <HEAD> <TITLE> <!-- WGITITLE --> </TITLE> <!-- WGBRANCH --> </HEAD> <BODY> <!-- WGINPUTFORM --> </BODY> </HTML> </pre>	<pre> <HTML> <HEAD> <TITLE> DECISION </TITLE> <SCRIPT LANGUAGE = "JavaScript "> <!-- function WBIBranch(form) { if (form.age.value < 20) location = "Decision3.html"; else if (form.age.value >= 65) location = "Decision4.html"; else location = "Decision5.html"; } //--> </Script> </HEAD> <BODY> <FORM METHOD = "get" NAME = "WBIForm1"> <TABLE> <TR> <TD> 나이 </TD> </TR> <TR> <TD> <INPUT TYPE = "text" NAME = "age"> 세 </TD> </TR> <TR> <TD> <INPUT TYPE = BUTTON VALUE = "확인" OnClick = "WBIBranch(this.form)"> </TD> </TR> </TABLE> </FORM> </BODY> </HTML> </pre>
---	--

<그림 15> 수치형 노드에 대한 디자인 템플레이트와 생성된 HTML 파일

트(<INPUT TYPE = "text" NAME = "age">)와 연결되어 있음에 유의하기 바란다. 또한 빈(null) 조건은 if문의 else로 처리되어 있는 점도 유의하기 바란다. 변수의 개수가 2개 이상인 경우에도 입력 부분에 INPUT 태그가 변수마다 추가되고, 클라이언트 측 스크립트 프로그램의 if문의 조건절에 여러 개의 변수들이 나타난다는 점을 제외하고는 위와 동일하게 처리할 수 있다. 이와 같은 방법으로 <그림 14>의 "2. 나이" 노드에 대하여 디자인 템플레이트 파일(<그림 15> 좌측)을 적용하여 생성한 HTML 파일과 그 디스플레이 화면이 <그림 15> 우측 및 <그림 16>에 나타나 있다.

현재 구현된 WeBIS는 스크립트 언어로써 클라이언트 측 스크립트 언어인 JavaScript외에도, 서버측 스크립트 언어를 사용하는 마이크로소프트 ASP(Active Server Pages) 프로그램도 생성할 수 있도록 되어 있으며, 또한 무선 인터넷을 위해 WML을 이용한 문서도 생성이 가능하다. 이외에도 데이터베이스 연결기능, 세션 기능, 기존 사이트 분석 기능, 화면 디자인 주제(Theme) 설정 기능 등 다양한 기능이 구현되어 있으나, 이러한 세세한 내용은 본 논문의 흐름을 벗어나는 것이므로 생략하도록 하겠다(<http://www.waisolutions.com/>).



<그림 16> 수치형 노드의 디스플레이 화면

VI. 결 론

지능형 웹 사이트의 필요성이 중요시 되면서 웹 기반 전문가 시스템 및 그 개발도구도 다시금 각광을 받고 있다. 기존의 웹 기반 전문가 시스템은 대부분 웹 시대 이전의 Stand-alone 방식의 전문가 시스템들을 CGI를 이용하여 웹 환경에 적용시킨 시스템들이다. 이들이 CGI의 문제점을 극복하기 위하여 Multi-thread, 트랜잭션 처리 등의 기법을 사용하여 대규모 서비스에 대비하고는 있으나, 전문가 시스템의 복잡성에 따른 무거운 프로세스(heavy-weight process)의 문제로부터 완전히 자유로운 것은 아니다. 이에 비하여, HTML 기반 추론은 트랜잭션 처리, 사실베이스의 유지 등에 따르는 웹 서버 부하의 문제와 "뒤로 가기" 버튼에 의한 사실베이스 혼란의 문제 등을 해결하였다. 그러나, HTML 기반 추론은 서로 하이퍼링크된 수 많은 HTML 파일들의 조합에 의해 이루어지기 때문에 이들을 생성하고 관리하는 것이 큰 과제가 된다. 이 문제를 해결하기 위하여 본 연구에서는 HTML 기반 추론용 웹 사이트를 생성하고 관리하기 위한 방법론을 제시하고, 이를 바탕으로 HTML 기반 전문가 시스템 개발 도구로서 WeBIS를 구현하였다.

WeBIS는 지식분석도라는 일종의 의사결정그래프를 GUI(Graphical User Interface) 환경에서 입력하도록 함으로써 지식의 입력 및 관리를 용이하게 하였으며, 이렇게 입력된 지식분석도와 화면 디자인을 표현한 디자인 템플레이트들을 조합하여 서로 하이퍼링크 된 HTML 파일들을 자동 생성한다. 생성된 HTML 파일들을 웹 서버에 복사함으로써 웹에 등록하면 웹을 통한 추론을 제공할 수 있다. 지식이 변경된 경우 지식공학자는 WeBIS에서 지식분석도를 수정한 후 HTML 파일들을 재생성한 후 생성된 HTML 파일들을 다시 웹 서버에 복사하면 되므로 사이트의 관리가 매우 용이하게 이루어질 수 있다.

지식의 입력 및 웹 사이트의 생성이라는 기본 기능 외에 주제(Theme) 설정에 의한 디자인 지원 기능, 데이터베이스의 값을 이용할 수 있도록 하는 데이터베이스 지원 기능, 사용자 인증 및 보안을 위한 세션 처리 기능 등 다양한 부가 기능이 실용적 측면에서 필요하며, 또 실제로 WeBIS에 구현되어 있다. 그리고, 기존의 추론 사이트의 유지 보수를 위하여 기존 사이트를 분석하여 지식분석도를 자동 생성하는 기능이 구현되어 있으며, 지식공학자는 이를 통하여 기존 사이트의 구조를 자동으로 지식분석도로 표현해 본 후 지식분석도의 수정 및 웹 사이트 재생성을 통하여 사이트를 관리할 수도 있다.

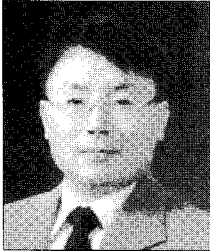
지식분석도의 선택형 노드는 웹 페이지에서 메뉴로 나타나며, 수치형 노드는 입력 박스로 나타나는 점을 다른 관점에서 생각해 보면, 지식분석도는 추론 사이트뿐만 아니라 일반적인 모든 웹 사이트에서 메뉴 구성 및 검색어 입력 등을 표현하기 위한 용도로도 사용될 수 있음을 알 수 있다. 따라서, WeBIS를 웹 사이트 개발 및 관리를 위한 CASE(Computer-aided Software Engineering) 도구로도 발전시킬 수 있을 것으로 생각한다.

〈참 고 문 헌〉

- [1] 김평철, "웹을 위한 데이터베이스 통로의 분류체계," WWW 96-1, 1996, pp. 50-66.
- [2] 송용욱, 이재규, "웹 기반 전문가 시스템의 자동생성 체계," 한국지능정보시스템학회 논문지, 제6권, 제1호, 2000, pp. 1-16.
- [3] 이재규, 송용욱, 권순범, 김우주, 김민용, UNIK를 이용한 전문가 시스템의 개발, 법영사, 1996A.
- [4] 이재규, 최형립, 김현수, 서민수, 주식진, 지원철, 전문가 시스템 원리와 개발, 법영사, 1996B.
- [5] 임규건, 강주영, 이재규, "Web 기반 전문가 시스템의 구조 분석," 한국전문가 시스템학회 '97 추계학술대회, pp. 63-73.
- [6] 최환진 역, Java 서블릿 프로그래밍, 한빛미디어, 1999.
- [7] Danesh, Arman, *Teach Yourself JavaScript in a Week*, Sams.net Publishing, 1996.
- [8] Dwight, Jeffrey, Michael Erwin, and Robert Niles, *Special Edition Using CGI*, Second Edition, QUE, 1997.
- [9] EXSYS, Inc., *Moving an EXSYS Application to the EXSYS Web Runtime Engine (WREN)*.
- [10] Eriksson, Henrik, "Expert Systems as Knowledge Servers," *IEEE Expert*, June 1996, pp. 14-19.
- [11] Far, Behrouz H. and Zenya Koono, "EXW-Pert System: A Web-Based Distributed Expert System for Groupware Design," *Expert Systems With Applications*, Vol. 11, No. 4, 1996, pp. 475-480.
- [12] Fielding, R., J. Gettys, J.C. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, *Hypertext Transfer Protocol - HTTP/1.1*, INTERNET-DRAFT <draft-ietf-http-v11-spec-rev-05>, Internet Engineering Task Force, 11 September 1998.
- [13] Giarratano, Joseph and Gary Riley, *Expert Systems: Principles and Programming*, 2nd Edition, PWS Publishing Company, 1994.
- [14] Jerke, N., M. Hatmaker, J. Anderson, *VBScript Interactive Course*, Waite Group Press, 1997.
- [15] Lee, J.K., I.K. Lee, and H.R. Choi, "Automatic rule generation by the transformation of Expert's Diagram: LIFT," *Int. J. Man-Machine Studies*, 32, 1990, pp. 275-292.
- [16] Lemay, Laura and Rogers Cadenhead, *Teach Yourself JAVA 1.2 in 21 Days*, SAMS Publishing, 1998.
- [17] O'Leary, Daniel E., "The Internet, Intranets, and the AI Renaissance," *IEEE Computer*, January 1997, pp. 71-78.
- [18] Raggett, Dave, Arnaud Le Hors, and Ian Jacobs(ed.), *HTML 4.0 Specification REC-html40-19980424*, W3C, 24 April 1998.
- [19] "Intelligent Internet Systems: Tools and Applications Part I," *Intelligent Software Strategies*, December 1996.
- [20] "Intelligent Internet Systems: Tools and Applications Part II," *Intelligent Software Strategies*, Winter 1997.
- [21] "Java Expert Systems Tools," *Intelligent Software Strategies*, Summer 1997.
- [22] "SELECTICA: Java-based Configuration for Internet and Electronic Commerce Applications," *Intelligent Software Strategies*, October 1996.

- [23] Amzi! URL <http://www.amzi.com/>.
- [24] Applied Logic Systems URL <http://www.als.com/>.
- [25] Attar Software URL <http://www.attar.com/>.
- [26] Blaze Advisor URL <http://www.blaze-soft.com/>.
- [27] Bowne Internet URL <http://www.bowne-internet.com/>.
- [28] Brightware URL <http://www.brightware.com/>.
- [29] Exsys URL <http://www.exsys.com/>.
- [30] Gensym URL <http://www.gensym.com/>.
- [31] ILOG Inc. URL <http://www.ilog.com/>.
- [32] Inference URL <http://www.inference.com/>.
- [33] IntelliSystems URL <http://www.intelli-systems.com/>.
- [34] JESS URL <http://herzberg1.ca.sandia.gov/jess/>.
- [35] MultiLogic URL <http://www.multilogic.com/>.
- [36] Neuron Data URL <http://www.neuron-data.com/>.
- [37] RadNet URL <http://www.radnet.com/>.
- [38] Selectica URL <http://www.selectica.com/>.
- [39] The Molloy Group URL <http://www.molloy.com/home.html>.
- [40] WAI Solutions URL <http://www.waisolutions.com/>.

◆ 저자소개 ◆



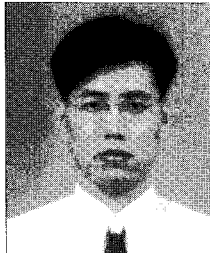
송용욱 (Song, Yong Uk)

1988년 서울대학교 국제경제학과를 졸업하고, 한국과학기술원에서 1990년도와 1995년도에 각각 석사 및 박사학위를 취득하였으며, 현재 연세대학교 원주캠퍼스 경영·정보학부 조교수로 재직하고 있다. 그의 연구분야는 전자상거래, 전자결제 및 보안, 경영분야 문제의 전문가 시스템 응용, 전문가 시스템 및 수리계획법과 전자상거래의 통합 등이다. 그의 연구 논문들은 Management Science, Annals of Operations Research 등에 게재되었다.



김우주 (Kim, Wooju)

연세대학교 상경대학 경영학과를 졸업하고, 한국과학기술원(KAIST) 경영과학과에서 경영정보학 석사 및 박사를 취득하였으며 현재 전북대학교 산업정보시스템공학과 부교수로 재직하고 있다. 최근의 주요 연구 관심 분야는 지능형 웹 (Intelligent Web)과 에이전트 기반 마켓플레이스 등을 중심으로 한 전자상거래 관련 분야와 데이터마이닝 및 웹 마이닝, 지식 관리 등이다. Decision Support Systems, Electronic Commerce Research and Applications, Expert Systems with Applications, New Review of Applied Expert Systems, AI Magazine, LNAI 등의 국제학술지를 포함하여 약 70여 편의 학술 논문, 학술 발표 및 저서 실적과 25건의 연구 과제 실적을 보이고 있다.



홍준석 (Hong, June Seok)

1989년에 서울대학교 경영학과를 졸업하고, 한국과학기술원에서 1991년도와 1997년도에 각각 석사 및 박사학위를 취득하였으며, 현재 경기대학교 경영정보학전공 조교수로 재직하고 있다. 그의 주된 연구분야는 전자상거래에서의 지능형 에이전트, 전자적자원관리(ERP), 경영의사결정을 위한 지능정보시스템, 전문가 시스템의 유지보수 등이다. 그의 연구 논문들은 Expert Systems with Applications, European Journal of Operational Research 등에 게재되었다.

◆ 이 논문은 2002년 10월 21일 접수하여 1차 수정을 거쳐 2003년 1월 9일 게재 확정되었습니다.